HMFEv - An Efficient Multivariate Signature Scheme

Albrecht Petzoldt[1(✉)], Ming-Shing Chen[2], Jintai Ding[3], and Bo-Yin Yang[2]

[1] National Institute for Standards and Technology, Gaithersburg, MD, USA
albrecht.petzoldt@nist.gov
[2] Academia Sinica, Taipei, Taiwan
{mschen,by}@crypto.tw
[3] University of Cincinnati, Ohio, USA
jintai.ding@gmail.com

Abstract. Multivariate Cryptography, as one of the main candidates for establishing post-quantum cryptosystems, provides strong, efficient and well-understood digital signature schemes such as UOV, Rainbow, and Gui. While Gui provides very short signatures, it is, for efficiency reasons, restricted to very small finite fields, which makes it hard to scale it to higher levels of security and leads to large key sizes.

In this paper we propose a signature scheme called HMFEv ("Hidden Medium Field Equations"), which can be seen as a multivariate version of HFEv. We obtain our scheme by applying the Vinegar Variation to the MultiHFE encryption scheme of Chen et al. We show both theoretically and by experiments that our new scheme is secure against direct and Rank attacks. In contrast to other schemes of the HFE family such as Gui, HMFEv can be defined over arbitrary base fields and therefore is much more efficient in terms of both performance and memory requirements. Our scheme is therefore a good candidate for the upcoming standardization of post-quantum signature schemes.

Keywords: Post-quantum cryptography · Multivariate cryptography · Signature schemes · NIST call for proposals

1 Introduction

Multivariate Public Key Cryptosystems (MPKCs) are one of the main candidates for guaranteeing the security of communication in a quantum world [1]. Multivariate schemes are in general very fast and require only modest computational resources, which makes them attractive for the use on low cost devices like smart cards and RFID chips [3,5]. Additionally, at least in the area of digital signatures, there exists a large number of practical multivariate schemes.

The existing multivariate signature schemes can be divided into two main groups. The first are the SingleField schemes such as UOV [15] and Rainbow [11], which follow the same type of design strategy using Oil-Vinegar polynomials. We

T. Lange and T. Takagi (Eds.): PQCrypto 2017, LNCS 10346, pp. 205–223, 2017.
DOI: 10.1007/978-3-319-59879-6_12

believe that these two schemes are more or less the best which can be achieved from this fundamental design.

On the other hand, we have the BigField schemes HFEv- [17] and Gui [18], which combine the HFE design with the Minus and Vinegar modifiers. These schemes make use of an HFE polynomial, whose degree D is very much affected by the size of the underlying field. We believe that, for security reasons, this degree should be chosen at least q^2+1, where q is the cardinality of the underlying field. However, during the signature generation process, we have to invert this univariate HFE polynomial and the complexity of this step can be estimated by $\mathcal{O}(D^3)$. To solve this conflict between security and efficiency, we have to build the scheme over very small finite fields such as GF(2) and GF(4). However, in this case, we have to choose the number of variables to be large, which leads to large key sizes and makes the scheme less efficient. Therefore it is a natural question, if it is possible to use large base fields such as GF(31) or GF(256) for the design of multivariate signature schemes of the HFEv- type.

In 2008, Chen et al. proposed a multivariate encryption scheme called MultiHFE [6], which can be seen as a multivariate version of HFE. While the scheme is very efficient, its security appeared to be weak and it was broken by Bettale et al. [2] by a generalization of the Kipnis-Shamir attack against HFE using the MinRank property of the system.

In this paper, we propose a signature scheme called HMFEv ("Hidden Medium Field Equations"), which we obtain by applying the Vinegar modification to MultiHFE. We show both theoretically and by experiments that our scheme is secure against direct and Rank attacks of the Kipnis-Shamir/Bettale type and analyze the security of our scheme against other known attacks against multivariate schemes, including differential attacks [7] and Hashimotos attack against the MultiHFE encryption scheme [14].

Our scheme can be seen as an extension of the Gui and QUARTZ signature schemes. However, by enabling a flexible choice of the base field, our new scheme overcomes a fundamental practical problem in the HFEv- design. While Gui and QUARTZ are, for efficiency reasons, mainly restricted to the field GF(2), our scheme allows the choice of an arbitrary base field. This allows us to reduce the number of equations and variables in the public system significantly, which leads to smaller key sizes and more efficient signature generation and verification processes. Furthermore, this enables an easy scalability of our scheme to higher levels of security. Our scheme is therefore a very strong candidate for the upcoming standardization of post-quantum signature schemes.

The rest of this paper is organized as follows. Section 2 gives an overview of the basic concepts of multivariate cryptography. In Sect. 3 we describe the MultiHFE encryption scheme which is the basis of our construction and analyze its security and efficiency. Section 4 describes our new HMFEv signature scheme in detail. In Sect. 5 we analyze the security of our scheme, in particular its behavior against direct and rank attacks. Section 6 proposes concrete parameter sets for our scheme for different levels of security, while Sect. 7 compares our HMFEv scheme with other multivariate signature schemes such as Gui and Rainbow. In

Sect. 8 we provide implementation details of our scheme and present performance results, and Sect. 9 concludes the paper.

2 Multivariate Cryptography

The public key of a multivariate public key cryptosystem (MPKC) is a set of multivariate quadratic polynomials. The security of these schemes is based on the MQ Problem of solving such a system. The MQ problem is proven to be NP-hard even for quadratic polynomials over the field GF(2) [13] and (for $m \approx n$) believed to be hard on average (both for classical and quantum computers).

To build a public key cryptosystem based on the MQ problem, one starts with an easily invertible quadratic map $\mathcal{F} : \mathbb{F}^n \to \mathbb{F}^m$ (central map). To hide the structure of $\mathcal{F}$ in the public key, one composes it with two invertible affine (or linear) maps $\mathcal{S} : \mathbb{F}^m \to \mathbb{F}^m$ and $\mathcal{T} : \mathbb{F}^n \to \mathbb{F}^n$. The *public key* of the scheme is therefore given by $\mathcal{P} = \mathcal{S} \circ \mathcal{F} \circ \mathcal{T} : \mathbb{F}^n \to \mathbb{F}^m$. The *private key* consists of $\mathcal{S}$, $\mathcal{F}$ and $\mathcal{T}$ and therefore allows to invert the public key.

In this paper we concentrate on multivariate schemes of the MediumField family. For this type of schemes, one chooses two integers k and ℓ and sets $n = k \cdot \ell$. The central map $\mathcal{F}$ of the scheme is a specially chosen easily invertible polynomial map over the vector space $\mathbb{E}^k$, where $\mathbb{E}$ is a degree ℓ extension field of $\mathbb{F}$. Using an isomorphism $\phi : \mathbb{F}^\ell \to \mathbb{E}$ we can transform $\mathcal{F}$ into a map

$$\bar{\mathcal{F}} = \underbrace{(\phi^{-1} \times \cdots \times \phi^{-1})}_{k-\text{times}} \circ \mathcal{F} \circ \underbrace{(\phi \times \cdots \times \phi)}_{k-\text{times}} : \mathbb{F}^n \to \mathbb{F}^n. \tag{1}$$

from $\mathbb{F}^n$ to itself. The map $\mathcal{F}$ is chosen in such a way that the map $\bar{\mathcal{F}}$ consists of multivariate quadratic polynomials. The *public key* has the form $\mathcal{P} = \mathcal{S} \circ \bar{\mathcal{F}} \circ \mathcal{T}$ with two invertible affine maps $\mathcal{S}, \mathcal{T} : \mathbb{F}^n \to \mathbb{F}^n$, the *private key* consists of $\mathcal{S}, \mathcal{F}$ and $\mathcal{T}$.

When the map $\mathcal{F}$ is bijective, the resulting scheme can be used both for signatures and public key encryption. The standard process of signature generation/verification respectively encryption/decryption works as shown in Fig. 1.

Signature Generation: To generate a signature for a document d, one uses a hash function $\mathcal{H} : \{0,1\}^\star \to \mathbb{F}^n$ to compute a hash value $\mathbf{w} = \mathcal{H}(d) \in \mathbb{F}^n$. After that, one computes recursively $\mathbf{x} = \mathcal{S}^{-1}(\mathbf{w}) \in \mathbb{F}^n$, $\mathbf{X} = (\phi \times \ldots \times \phi)(\mathbf{x}) \in \mathbb{E}^k$, $\mathbf{Y} = \mathcal{F}^{-1}(\mathbf{X}) \in \mathbb{E}^k$, $\mathbf{y} = (\phi^{-1} \times \ldots \times \phi^{-1})(\mathbf{Y}) \in \mathbb{F}^n$ and $\mathbf{z} = \mathcal{T}^{-1}(\mathbf{y})$. The signature of the document d is $\mathbf{z} \in \mathbb{F}^n$.

Signature Verification: To check, if $\mathbf{z} \in \mathbb{F}^n$ is indeed a valid signature for the document d, one computes the hash value $\mathbf{w} = \mathcal{H}(d) \in \mathbb{F}^n$ and $\mathbf{w}' = \mathcal{P}(\mathbf{z}) \in \mathbb{F}^n$. If $\mathbf{w}' = \mathbf{w}$ holds, the signature is accepted, otherwise rejected.

A good overview of existing multivariate schemes can be found in [8].

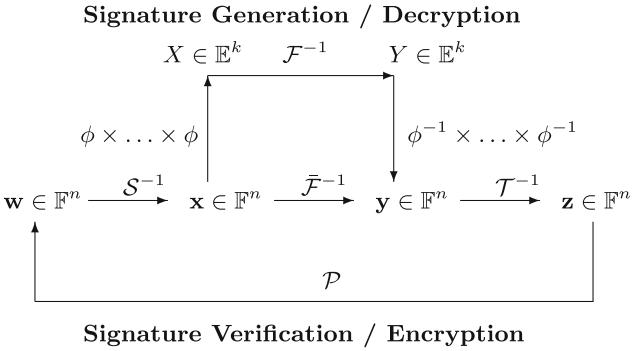

Fig. 1. Workflow of multivariate MediumField schemes

3 The MultiHFE Scheme

An important example for a multivariate scheme from the MediumField family is the MultiHFE scheme of Chen et al. [6]. In its basic version, the scheme can be used both as an encryption and signature scheme.

The k components $\mathcal{F}^{(1)}, \ldots, \mathcal{F}^{(k)}$ of the central map $\mathcal{F}$ are of the form

$$\mathcal{F}^{(i)} = \sum_{r=1}^{k} \sum_{s=r}^{k} \alpha_{rs}^{(i)} \cdot X_r X_s + \sum_{r=1}^{k} \beta_r^{(i)} \cdot X_r + \gamma^{(i)} \ (i = 1, \ldots, k) \tag{2}$$

with coefficients $\alpha_{rs}^{(i)}$, $\beta_r^{(i)}$ and $\gamma^{(i)}$ randomly chosen from $\mathbb{E}$. Note that the polynomials $\mathcal{F}^{(i)}(i = 1, \ldots, k)$ are multivariate polynomials of the HFE type with $D = 2$. The map $\bar{\mathcal{F}}$ of the MultiHFE signature scheme is defined as shown in Eq. (1) and is, due to the Frobenius isomorphism, a multivariate quadratic map over the vector space $\mathbb{F}^n$. To hide the structure of $\bar{\mathcal{F}}$ in the public key, one composes it with two invertible affine maps $\mathcal{S}$ and $\mathcal{T} : \mathbb{F}^n \to \mathbb{F}^n$. Therefore, the *public key* of the scheme is given by $\mathcal{P} = \mathcal{S} \circ \bar{\mathcal{F}} \circ \mathcal{T} : \mathbb{F}^n \to \mathbb{F}^n$, the *private key* consists of $\mathcal{S}$, $\mathcal{F}$ and $\mathcal{T}$.

Signature Generation: In order to generate a signature for a message d, one uses a hash function $\mathcal{H} : \{0,1\}^\star \to \mathbb{F}^n$ to compute the hash value $\mathbf{w} = \mathcal{H}(d) \in \mathbb{F}^n$ and performs the following three steps.

1. Compute $\mathbf{x} = \mathcal{S}^{-1}(\mathbf{w}) \in \mathbb{F}^n$ and lift the result to the vector space $\mathbb{E}^k$. Denote the result by $\mathbf{X}$.
2. Invert the central map $\mathcal{F}$ to obtain $\mathbf{Y} = \mathcal{F}^{-1}(\mathbf{X}) \in \mathbb{E}^k$ and compute $\mathbf{y} = (\phi^{-1} \times \ldots \times \phi^{-1})(\mathbf{Y}) \in \mathbb{F}^n$. Since $\mathcal{F}$ is a system of k randomly chosen quadratic polynomials in k variables, we need for this step a system solver like XL [20] or a Gröbner Basis algorithm such as F_4 [12] or F_5.
3. Compute the signature $\mathbf{z} \in \mathbb{F}^n$ by $\mathbf{z} = \mathcal{T}^{-1}(\mathbf{y})$.

Signature Verification: To check, if $\mathbf{z} \in \mathbb{F}^n$ is indeed a valid signature for a message d, one computes the hash value $\mathbf{w} = \mathcal{H}(d)$ and $\mathbf{w}' = \mathcal{P}(\mathbf{z})$. If $\mathbf{w}' = \mathbf{w}$ holds, the signature is accepted, otherwise rejected.

3.1 Efficiency

The most complex step during the signing process of MultiHFE is the solution of the multivariate quadratic system $\mathcal{F}(Y_1, \ldots, Y_k) = (X_1, \ldots, X_k)$ (k equations in k variables) over the extension field $\mathbb{E}$. Since the coefficients of the system $\mathcal{F}$ are random elements of $\mathbb{E}$, we need for this step a system solver like XL [20] or a Gröbner Basis algorithm such as F_4 [12]. If the number k of equations and variables in this system is small, these algorithms can invert $\mathcal{F}$ very efficiently. However, when the parameter k gets larger, the decryption process of MultiHFE becomes very costly and the scheme therefore gets inefficient.

3.2 The Rank Attack Against HFE and MultiHFE

In [16], Kipnis and Shamir proposed a rank based attack against the univariate HFE scheme. The key idea of this attack is to lift all the maps $\mathcal{S}$, $\mathcal{P}$ and $\mathcal{T}$ to univariate maps $\mathcal{S}^\star$, $\mathcal{P}^\star$ and $\mathcal{T}^\star$ over the extension field $\mathbb{E}$. Since the rank of the central map $\mathcal{F}$ is bounded from above by $r = \lfloor \log_q(D-1) \rfloor + 1$, this enabled them to recover the private key by solving an instance of a MinRank problem. However, since computing the map $\mathcal{P}^\star$ appeared to be very costly, the attack of Kipnis and Shamir is not very efficient.

Later, Bettale et al. [2] found a way to perform the attack of Kipnis and Shamir without the need of recovering the map $\mathcal{P}^\star$. Besides improving the efficiency of the Kipnis-Shamir attack, this makes it much easier to extend the attack to MultiHFE. Due to lack of space we cannot present all the details of the attacks of Kipnis-Shamir and Bettale here and refer to the papers [2,16] for a detailed description of the attacks. Here, we just present the main results of [2].

Theorem 1. *For MultiHFE, recovering the affine transformation $\mathcal{T}$ reduces to simultaneously solving k MinRank problems over the base field.*

With this, Bettale et al. could further prove

Theorem 2. *The complexity of solving the MultiHFE MinRank problem is $\mathcal{O}(\ell^{(k+1)\omega})$ with $2 < \omega \leq 3$ being the linear algebra constant and ℓ being the degree of the field extension $\mathbb{E}|\mathbb{F}$.*

We therefore face the following problem: If the parameter k in MultiHFE is small, the scheme can be easily broken by the MinRank attack. On the other hand, if we choose k larger, the efficiency of the scheme becomes quite bad.

In the following, we show how to solve this dilemma by modifying the MultiHFE scheme.

4 The New Signature Scheme HMFEv

Let $\mathbb{F}$ be a finite field and k, ℓ and v be integers. We set $n = k \cdot \ell$. Furthermore, let $g(X) \in \mathbb{F}[X]$ be an irreducible polynomial of degree ℓ and $\mathbb{E} = \mathbb{F}[X]/g(X)$ the corresponding extension field. We define an isomorphism $\phi : \mathbb{F}^\ell \rightarrow \mathbb{E}$ by

$$\phi(x_1, \ldots, x_\ell) = \sum_{i=1}^{\ell} x_i \cdot X^{i-1}.$$

The *central map* $\mathcal{F} : \mathbb{E}^k \times \mathbb{F}^v \rightarrow \mathbb{E}^k$ of the scheme consists of k components $\mathcal{F}^{(1)}, \ldots, \mathcal{F}^{(k)}$ of the form

$$\mathcal{F}^{(i)} = \sum_{r=1}^{k} \sum_{s=r}^{k} \alpha_{rs}^{(i)} \cdot X_r X_s + \sum_{r=1}^{k} \beta_r^{(i)}(v_1, \ldots, v_v) \cdot X_r + \gamma^{(i)}(v_1, \ldots, v_v) \qquad (3)$$

with coefficients $\alpha_{rs}^{(i)} \in_R \mathbb{E}$, linear functions $\beta_r^{(i)} : \mathbb{F}^v \rightarrow \mathbb{E}$ and quadratic maps $\gamma^{(i)} : \mathbb{F}^v \rightarrow \mathbb{E}$ ($i \in \{1, \ldots, k\}$).

Due to the special form of $\mathcal{F}$, the map

$$\bar{\mathcal{F}} = \underbrace{(\phi^{-1} \times \cdots \times \phi^{-1})}_{k-\text{times}} \circ \mathcal{F} \circ (\underbrace{\phi \times \cdots \times \phi}_{k-\text{times}} \times \mathrm{id}_v)$$

is a multivariate quadratic map from $\mathbb{F}^{n+v}$ to $\mathbb{F}^n$. Here, id_v is the identity map over the vector space $\mathbb{F}^v$.

To hide the structure of $\bar{\mathcal{F}}$ in the public key, we combine it with two randomly chosen invertible affine maps $\mathcal{S} : \mathbb{F}^n \rightarrow \mathbb{F}^n$ and $\mathcal{T} : \mathbb{F}^{n+v} \rightarrow \mathbb{F}^{n+v}$.

The *public key* of the scheme is given by

$$\mathcal{P} = \mathcal{S} \circ \bar{\mathcal{F}} \circ \mathcal{T} : \mathbb{F}^{n+v} \rightarrow \mathbb{F}^n,$$

the *private key* consists of $\mathcal{S}$, $\mathcal{F}$ and $\mathcal{T}$.

Signature Generation: To generate a signature for a document d, we use a hash function $\mathcal{H} : \{0,1\}^\star \rightarrow \mathbb{F}^n$ to compute the hash value $\mathbf{w} = \mathcal{H}(d) \in \mathbb{F}^n$. After that, we perform the following six steps

1. Compute $\mathbf{x} = \mathcal{S}^{-1}(\mathbf{w}) \in \mathbb{F}^n$.
2. Lift the result to the vector space $\mathbb{E}^k$ by computing $\mathbf{X} = (X_1, \ldots, X_k)$ with $X_i = \phi(x_{(i-1)\cdot\ell+1}, \ldots, x_{i\cdot\ell})$ $(i = 1, \ldots, k)$.
3. Choose random values for the Vinegar variables $v_1, \ldots, v_v \in \mathbb{F}$ and substitute them into the central map components to obtain the parametrized maps $\mathcal{F}_V^{(1)}, \ldots, \mathcal{F}_V^{(k)}$.
4. Use the XL-Algorithm or a Gröbner Basis method to compute $Y_1, \ldots, Y_k$ such that $\mathcal{F}_V^{(i)}(Y_1, \ldots, Y_k) = X_i (i = 1, \ldots, k)$.
5. Move the result down to the base field by computing $\mathbf{y} = (\phi^{-1}(Y_1), \ldots, \phi^{-1}(Y_k), v_1, \ldots, v_v) \in \mathbb{F}^{n+v}$.
6. Compute the signature $\mathbf{z} \in \mathbb{F}^{n+v}$ by $\mathbf{z} = \mathcal{T}^{-1}(\mathbf{y})$.

Signature Verification: In order to check, if $\mathbf{z} \in \mathbb{F}^{n+v}$ is indeed a valid signature for the document d, one computes $\mathbf{w} = \mathcal{H}(d)$ and $\mathbf{w}' = \mathcal{P}(\mathbf{z})$. If $\mathbf{w}' = \mathbf{w}$ holds, the signature is accepted, otherwise rejected.

5 Security

In this Section we analyze the security of our scheme. In particular we study both theoretically and using computer experiments the behavior of our scheme against direct and rank attacks.

5.1 Direct and Rank Attacks

The complexity of a direct attack is closely related to the degree of regularity of the system. Therefore the key task is to study the degree of regularity of a direct attack against our scheme.

From the work of Ding and Hodges in Crypto 2011 [10] we know that the degree of regularity of a direct attack against an HFE scheme can be estimated by looking at a single polynomial in the extension field $\mathbb{E}$, and the rank of the associated quadratic form.

In the case of HMFEv, the situation is slightly different, but still very similar. For HMFEv, the components of the public key come from several polynomials over the medium field, which are given as

$$\mathcal{F}^{(i)} = \sum_{r=1}^{k} \sum_{s=r}^{k} \alpha_{rs}^{(i)} \cdot X_r X_s + \sum_{r=1}^{k} \beta_r^{(i)}(v_1, \ldots, v_v) \cdot X_r + \gamma^{(i)}(v_1, \ldots, v_v) \; (1 \leq i \leq k).$$

Using the same argument as in the work of Ding and Yang in [9] we can, under the assumption of $v \leq \ell$, lift each map $\mathcal{F}^{(i)}$ $(1 \leq i \leq k)$, which is a map from $\mathbb{E}^k \times \mathbb{F}^v$ to $\mathbb{E}$, to a map $\mathcal{F}'^{(i)}$ from $\mathbb{E}^{k+1}$ to $\mathbb{E}$. Here, the additional component in the domain comes from the use of the Vinegar variables. Then we can look at the rank of the quadratic form associated to the polynomial $\mathcal{F}'^{(i)}$ as in the case of the original Kipnis-Shamir attack.

Using the same method as in [9] we can prove

Theorem 3. *If $v \leq \ell$ holds, the rank of the quadratic form associated to $\mathcal{F}'^{(i)}$ is less or equal to $k + v$.*

Proof **(sketch).** The main idea of the proof is to lift the central map back to a vector space of $k+1$ copies of $\mathbb{E}$, namely $\mathbb{E}^{k+1}$, where we will use the additional copy of $\mathbb{E}$ to accommodate the Vinegar variables. Then we can use the same analysis as in [9] to derive the proof.

Under the assumption that the Vinegar maps $\beta_r^{(i)}$ look like random functions, we find that the lower bound given by Theorem 3 is tight.

From this result we directly derive a lower bound for the complexity of the MinRank attack (see Theorem 2) by

$$\text{Complexity}_{\text{MinRank}} \geq \ell^{(k+v+1)\cdot\omega}. \tag{4}$$

Theorem 3 allows us to use the method of [10] to derive directly.

Theorem 4. *The degree of regularity of a direct attack against an HMFEv system is, under the assumption of $v \leq \ell$, upper bounded by*

$$d_{\text{reg}} \leq \begin{cases} \frac{(q-1)(k+v-1)}{2} + 2 & \text{if } q \text{ even and } (k+v) \text{ odd} \\ \frac{(q-1)\cdot(k+v)}{2} + 2 & \text{otherwise} \end{cases}. \tag{5}$$

Equation (5) gives an upper bound for the degree of regularity of a direct attack against our scheme. However, in order to estimate the security of the HMFEv scheme in practice, we need to analyze if the bound given by (5) is reasonably tight. Furthermore we want to study, if, as Eq. (5) indicates, only the sum and not the concrete choice of k and v determines the degree of regularity of a direct attack against an HMFEv system. To answer these two questions, we performed a large number of experiments.

Our experiments (see in Sect. A of the appendix of this paper) show that the upper bound on the degree of regularity given by Eq. (5) is relatively tight. We could find several MHFEv instances which actually meet the upper bound and found that in most other cases the upper bound is missed only by one. Regarding the second question, we found that the concrete choice of k and v has no influence on the behavior of the scheme against direct attacks as long as k and v are not too small.

The experiments in the appendix deal with HMFEv schemes over very small fields such as GF(2) and GF(3). However, one major benefit of the HMFEv scheme is that, in contrast to HFEv-, it can be efficiently used over larger fields, too. As our experiments (see Sect. 6) show, these systems behave much more like random systems and we can reach high degrees of regularity, by which we can show the security of our scheme against direct attacks.

5.2 Quantum Attacks

In [19], Schwabe and Westerbaan showed that a binary system of n multivariate quadratic equations can be solved by a quantum computer in time

$$\text{comp}_{\text{MQquantum;GF(2)}} = 2^{n/2} \cdot 2 \cdot n^3. \tag{6}$$

Since our systems over GF(256) can easily be translated into systems over GF(2), this attack affects also our scheme (at least in theory). However, since this transition increases the number of variables in the system by a factor of 8, it has no major effect on the parameter selection of our scheme.

5.3 Other Attacks and a Remark on the Minus Method

Additional to direct, quantum and rank attacks, we analyzed the security of our scheme against other known attacks against multivariate schemes, including differential attacks [7] and Hashimotos attack against the original MultiHFE encryption scheme [14]. Obviously, this attack is essentially a differential symmetry attack though it is not formulated in that way. Therefore it is important to perform a solid analysis of the differential attacks for the new scheme. However, all the recent work in differential attacks indicates that it is a very special attack that is applicable ONLY to very special systems with lowest possible rank. For our scheme, this is clearly not the case and the Vinegar variables destroy efficiently all differential symmetries [4]. However, the complete analysis is very tedious, and our analysis will be presented in a subsequent paper.

Remark. A natural question here is, why we do not use the Minus method as in the case of HFEv-. There are two main reasons.

1. In contrast to the Vinegar variation, the Minus modification does not help to defend our scheme against differential attacks and Hashimotos attack against the original MultiHFE encryption scheme [14].
2. If we apply the same matrix rank method used in the proof of Theorem 3 (see also [9]) to MHFEv- (i.e. we lift the central map back to a vector space of $k+1$ copies of $\mathbb{E}$, where we use the additional copy of $\mathbb{E}$ to accommodate the Minus parameters and Vinegar variables), this directly leads to the conclusion that the MinRank should be $k+v+ak$, where a is the number of Minus equations. If we follow the above method further, we derive

$$d_{\text{reg}} \leq \begin{cases} \frac{(q-1)(k+v+ak-1)}{2} + 2 & \text{if } q \text{ even and } (k+v+ak) \text{ odd} \\ \frac{(q-1)\cdot(k+v+ak)}{2} + 2 & \text{otherwise} \end{cases} . \quad (7)$$

   However our experiments show that this bound is not tight. This can be explained as follows. In the case of HFEv-, the estimate comes from using a single polynomial over the extension field, and a single polynomial already determines the whole system; in the case of MHFEv-, the system is determined by k polynomials, not by one; since our analysis considers only one of these polynomials, it does not use all the information available and therefore overestimates the degree of regularity.

This means we have a gap in the knowledge on estimating the degree of regularity in MHFEv-, which is the reason we propose the MHFEv system (i.e. only with Vinegar). This problem is very interesting and important, and we are going to deal with it in a subsequent paper.

6 Parameter Choice

In this section we consider the question how to find good parameter sets for our scheme. In particular, we aim at finding parameters for HMFEv over the fields GF(31) and GF(256).[1]

6.1 How to Choose the Parameter k?

The first question we have to answer in order to find suitable parameters for our scheme is how to choose the parameter k and therefore the number of components of the central map. Reducing the value of k will speed up the signature generation process of our scheme since it decreases the size of the multivariate quadratic system we have to solve. However, if k is too small, this might bring the security of our scheme into jeopardy.

For fields of odd characteristic (e.g. $\mathbb{F} = \mathrm{GF}(31)$) we choose the parameter k to be 2. However, in order to increase the security of our scheme against Rank attacks, we choose in this case the components of the central map $\mathcal{F}$ in a special way. Let F_1 and F_2 be the 2×2 matrices representing the homogeneous quadratic parts of the maps $\mathcal{F}^{(1)}$ and $\mathcal{F}^{(2)}$. A linear combination of F_1 and F_2 of rank 1 exists if and only if the quadratic polynomial $p(X) = \det(F_1 + X \cdot F_2) \in \mathbb{E}[X]$ has a solution. We therefore choose the coefficients of $\mathcal{F}^{(1)}$ and $\mathcal{F}^{(2)}$ in such a way that the polynomial $p(X)$ is irreducible.

For fields of even characteristic, the symmetric matrices representing the quadratic maps $\mathcal{F}^{(i)}$ contain zero elements on the main diagonal. Therefore, for $k = 2$, the rank of these matrices would be 1 and the upper linear combination of the maps $\mathcal{F}^{(1)}$ and $\mathcal{F}^{(2)}$ would actually lead to a matrix of rank 0 (i.e. no quadratic terms at all). To prevent this, we choose for fields of even characteristic the parameter k to be 3.

6.2 Experiments with Direct Attacks Against HMFEv Schemes over GF(31) and GF(256)

In Sect. 5.1 we already presented some results of experiments with the direct attack against HMFEv instances. However, in Sect. 5.1, we looked at HMFEv schemes over very small fields, for which the bound given by Eq. (5) is more or less tight. In this section we consider the question, if concrete instances of HMFEv over the larger fields GF(31) and GF(256) are hard to solve.

To do this, we created, for different parameter sets, HMFEv systems over GF(31) and GF(256) and solved these systems, after fixing v variables to obtain a determined system, with the F_4 algorithm integrated in MAGMA. The experimentswere performed on a single core of a server with 16 AMD Opteron

[1] The reason why we do not propose parameters for our scheme over GF(16) is the following: To defend the scheme against the quantum attack (see Sect. 5.2), we need a large number of equations over GF(16). This actually makes the schemes less efficient than HMFEv over GF(31) or GF(256).

Table 1. Experiments with the direct attack against HMFEv schemes over GF(31) and GF(256)

GF(31)	Parameters (k, ℓ, v)	$(2, 6, 4)$	$(2, 7, 4)$	$(2, 8, 4)$	Random
	m,n	12,12	14,14	16,16	16,16
	d_{reg}	14	16	18	18
	time (s)	1,911	164,089	-	-
	Memory (MB)	953	17,273	ooM	ooM
GF(256)	Parameters (k, ℓ, v)	$(3, 3, 6)$	$(3, 4, 6)$	$(3, 5, 6)$	Random
	m,n	9,9	12,12	15,15	15,15
	d_{reg}	11	14	17	17
	Time (s)	3.9	1,853	-	-
	Memory (MB)	23.7	952	ooM	ooM

processors (2.4 GHz) and 128 GB of RAM. For each parameter set we performed 10 experiments. Table 1 shows the results.

As the table shows, we can, for HMFEv instances over both GF(31) and GF(256), reach high degrees of regularity. In particular we see that, for the parameter sets proposed in the next section, the degree of regularity of a direct attack is at least 17. When solving the resulting linear systems with a sparse Wiedemann solver, we can estimate the complexity of a direct attack by

$$\text{Complexity}_{\text{direct attack}} \approx 3 \cdot \binom{n + d_{\text{reg}}}{d_{\text{reg}}}^2 \cdot \binom{n}{2}. \tag{8}$$

By substituting the value $d_{\text{reg}} = 17$ into this formula we find that the complexity of a direct attack against the HMFEv instances shown in Table 2 is beyond the claimed levels of security.

Also note that, for the underlying fields of GF(31) and GF(256), the public systems of HMFEv behave very similar to random systems. This also holds when guessing some variables before applying the F_4 algorithm (hybrid approach).

6.3 Parameters

Table 2 shows, for different levels of security (128, 192, and 256 bit) our parameter recommendations for the HMFEv signature scheme over GF(31) and GF(256). In the case of GF(31), we store one element of GF(31) in 5 bits, while 24 bits can be efficiently stored in 5 GF(31) elements.

The parameter sets given in Table 2 are chosen in such a way that the complexities of direct attacks (including hybrid approach; see Sect. 6.2), quantum attacks (see Eq. (6)) and Rank attacks (see Eq. (4)) against the given HMFEv instances are beyond the claimed levels of security. To be on the conservative side we chose, in formula (4), the linear algebra constant ω to be 2. Furthermore, in

Table 2. Parameter recommendations for the HMFEv signature scheme

Quantum security level (bit)	Parameters $(\mathbb{F}, k, \ell, v)$	Public key size (kB)	Private key size (kB)	Hash size (bit)	Signature size (bit)
128	(GF(31),2,28,12)	81.8	8.9	277	337
	(GF(256),3,15,16)	85.8	15.2	360	488
192	(GF(31),2,40,17)	234.7	20.0	396	481
	(GF(256),3,23,21)	282.1	35.0	552	720
256	(GF(31),2,55,21)	583.9	38.0	544	649
	(GF(256),3,31,26)	659.4	65.3	744	952

the case of MHFEv over GF(31), we had to take care of the fact that the public systems contain enough equations to prevent collision attacks against the hash function.

7 Comparison

The basic idea of the HMFEv signature scheme is very similar to that of Gui [18]: by applying the Vinegar modification it is possible to increase both the security and the efficiency of the scheme significantly. However, there are at least three major advantages of our scheme compared to Gui.

Key Sizes: First, for efficiency reasons, the Gui signature scheme is mainly restricted to the field GF(2). This leads to a large number of variables in the scheme and therefore to large key sizes. On the other hand, the HMFEv signature scheme can be defined over larger fields, too. This enables us to decrease the number of variables in the system and therefore reduces the public key size of the scheme significantly (see Table 3).

Simplicity and Efficiency: Secondly, for the parameter sets recommended in [18], the output size of the HFEv- public key is only 90 bit. Therefore, in order to defend the HFEv- signature scheme against collision attacks, the authors of Gui had to create a specially designed signature generation process for their scheme which inverts the HFEv- core map several times. Since the design of Gui requires the single HFEv- systems to have exactly one solution, generating one single Gui signature implies about 11 inversions of the HFEv- map, which leads to a relatively low performance of Gui. In the case of the HMFEv scheme, we do not need this multiple inversion of the core map, which makes the signature generation process of our scheme much faster and easier to implement. Furthermore, since the number of variables in the public systems of Gui is much larger than for our scheme, the evaluation of the HMFEv public systems and therefore the verification process of our scheme is much cheaper. This advantage of our scheme is increased by the fact that, during the verification process of Gui, we have to evaluate the public system several times.

Scalability: The third major advantage of the HMFEv scheme is that, in contrast to other HFEv- based schemes like Gui, the scheme can be scaled much

easier to higher levels of security. For example, in order to obtain a quantum security level of 256 bit, we need an internal state of at least 457 bit (see Eq. (6)), which means that we need at least 457 variables over GF(2). This would lead to key sizes which are completely impractical (see Table 3). In the case of HMFEv, the necessary increase of the number of variables is far less drastical. Alternatively, we can increase the size of the internal state simply by choosing a larger base field. For both strategies, the resulting increase of the key size is far less significant.

Table 3 compares, for different levels of security, the HMFEv and Gui signature schemes with respect to key and signature sizes. Note here that the parameters proposed in [18] are not chosen to provide quantum security. In order to provide a fair comparison, we therefore extrapolated the parameters of [18] to meet the quantum security levels. For better comparison, Table 3 also shows key and signature sizes of Rainbow.

Table 3. Comparison of our scheme with other multivariate signature schemes

Quantum security level (bit)		Public key size (kB)	Private key size (kB)	Signature size (bit)
80	Rainbow (GF(256),17,13,13)	25.1	19.9	344
	Gui (GF(2),120,9,3,3,2)	110.7	3.8	129
	HMFEv (GF(31),2,18,8)	**22.5**	**3.5**	**218**
	HMFEv (GF(256),3,9,12)	**21.6**	**6.0**	**312**
128	Rainbow (GF(256),36,21,22)	136.0	102.5	632
	Gui (GF(2),212,9,3,4,2)	592.8	11.6	222
	HMFEv (GF(31),2,28,12)	**81.8**	**8.9**	**337**
	HMFEv (GF(256),3,15,16)	**85.8**	**15.2**	**488**
256	Rainbow (GF(256),86,45,46)	1,415.7	1,046.3	1,416
	Gui (GF(2),464,9,7,8,2)	6,253.7	56.4	488
	HMFEv (GF(31),2,55,21)	**583.9**	**38.0**	**649**
	HMFEv (GF(256),3,31,26)	**659.4**	**65.3**	**952**

As Table 3 shows, the key sizes of our scheme are much smaller than that of Rainbow and Gui (especially for high levels of (quantum) security). The reason for this is that our scheme combines the main advantages of Rainbow and Gui: similar to Rainbow, we can use the HMFEv scheme over large finite fields, which reduces the number of equations needed in the public system. Similar to Gui, our scheme has a small blow up factor between the number of equations and the number of variables of about 1.25 (for Rainbow, this factor is about 1.8). This reduces both key and signature sizes significantly.

8 Implementation

In this section we provide some implementation details for our scheme and present performance results. In particular, we describe here

– how to efficiently invert the central map $\mathcal{F}$ and
– how to perform arithmetic operations in $\mathbb{F}$ and $\mathbb{E}$ efficiently.

8.1 Inversion of the Central Map $\mathcal{F}$

The most costly step during the signature generation process of our scheme is the inversion of the central equation $\mathcal{F}_V(\mathbf{Y}) = \mathbf{X}$, which is given as a system of k multivariate quadratic equations in k variables over the extension field $\mathbb{E}$. Since the coefficients of this system are random $\mathbb{E}$-elements, we need a system solver such as XL or a Gröbner Basis algorithm for this step.

Obviously, the complexity of solving the system $\mathcal{F}_V(\mathbf{Y}) = \mathbf{X}$ and therefore the complexity of the signature generation process depends mainly on the choice of the parameter k. A small value of k will reduce the number of $\mathbb{E}$-multiplications in this process. However, it also leads to large extension fields and therefore increases the cost of a single $\mathbb{E}$-multiplication. Furthermore, choosing k too small might weaken the security of our scheme (see Sect. 6.1).

To find the optimal parameter k for our scheme, we therefore have to analyze the process of inverting the central map $\mathcal{F}_V$ in more detail. Let the multivariate system $\mathcal{F}_V$ be given by the k multivariate quadratic maps $\mathcal{F}_V^{(1)}, \ldots, \mathcal{F}_V^{(k)} : \mathbb{E}^k \to \mathbb{E}$. As we find, the process of solving the multivariate system $\mathcal{F}_V(\mathbf{Y}) = \mathbf{X}$ consists mainly of two parts:

1. (**Gröbner Basis step**) Find a univariate polynomial $p : \mathbb{E} \to \mathbb{E}$ in the ideal $\langle \mathcal{F}_V^{(1)}, \ldots, \mathcal{F}_V^{(k)} \rangle$.
2. (**Solving step**) Solve the polynomial p by Berlekamp's algorithm.

In the following we analyze, for different values of k, these two steps in detail. For this, we fix the number $n = k \cdot \ell$ to $n = 48$ and choose $k \in \{2, 3, 4\}$. Inverting the system $\mathcal{F}_V$ therefore relates to

– solving a system of 2 quadratic equations in 2 variables over $\mathbb{F}^{24}$ or
– solving a system of 3 quadratic equations in 3 variables over $\mathbb{F}^{16}$ or
– solving a system of 4 quadratic equations in 4 variables over $\mathbb{F}^{12}$.

For $k = 2, 3$, we use for the first part a specially designed Gröbner Basis method tailored for the occasion. In the case of 2 quadratic equations in 2 variables, we run in the Gröbner Basis step successively 2 Gaussian eliminations on matrices of size 5×9 and 7×10. By doing so, we obtain a single variable equation p of degree 4. To perform this step, we need about $5 \cdot (11 + 12) + 7 \cdot 8 \cdot 4 = 339$ multiplications over the field $\mathbb{F}^{24}$.

In the Solving step, we have to solve the univariate equation p of degree 4 over the field $\mathbb{F}^{24}$. This takes about $6 \cdot 4^2 \cdot 24 = 2{,}304$ multiplications over the field $\mathbb{F}^{24}$. One can see that the overall complexity is dominated by the Solving step.

In the case of 3 quadratic equations in 3 variables, we run in the Gröbner Basis step successively 3 Gaussian eliminations on matrices of size 11×19, 8×16

and 5×13 with many zero elements to derive a single variable equation of degree 8. For this we need about 1,700 $\mathbb{F}^{16}$-multiplications.

Then we solve this single variable equation of degree 8 over $\mathbb{F}^{16}$. This requires about $6 \cdot 8^2 \cdot 16 = 6{,}144$ $\mathbb{F}^{16}$-multiplications. One can see that the Solving step again dominates the complexity.

In the case of 4 quadratic equations in 4 variables, the situation is too complicated to do it by hand and we use the F_4 algorithm directly. In this case, we run successively Gaussian eliminations on matrices of size $19 \times 34, 41 \times 50, 42 \times 50$ and 35×48, which requires about $2 \cdot 50^3 = 250{,}000$ $\mathbb{F}^{12}$ multiplications. By doing so, we obtain a single variable equation p of degree 16.

In the Solving step, we have to solve this univariate equation p over the field $\mathbb{F}^{12}$, which requires about $6 \cdot 16^2 \cdot 12 = 18{,}432$ multiplications. One can see that here the solving of the single variable equation does not dominate the complexity anymore.

8.2 Arithmetic over Finite Fields

Evaluating the public map requires first to generate all monomials, and then the computation of the inner products between coefficient and monomial vectors. The first step requires $n(n+1)/2$ field multiplications. The second part is much more important and requires $mn(n+3)/2$ multiplications in the base field and nearly as many additions (or XORs) to accumulate the results.

Arithmetic in GF(256) is done via the table-lookup instruction VPSHUFB. This instruction allows 32 simultaneous lookups from a table of 16 elements, which allows for easy scalar-vector multiplications of GF(16) elements using log-exp tables. Every 32 GF(16) multiplications then take two VPSHUFB instructions and an add in addition to the required VPXOR, since we store the public key in log form. Finally we put together multiplications of GF(256) for the public key using four multiplications in GF(16) (schoolbook method).

The main computation in big binary fields uses PCLMULQDQ and schoolbook methods, because on recent processors this instruction is really fast. We also use lazy reductions, which means that we often do not reduce to the lowest degree. A time-constant complete reduction is performed after the entire operation.

Arithmetics in GF(31) use AVX2 instructions (and following that SSSE3 instructions). For best use of our resources, we use a YMM register to represent a vector of 16 or 32 coefficients in the public key to be multiplied by two monomials. Values for two monomials each time are also expanded into an YMM register. The actual arithmetic uses the VPMADDUSBW instruction to multiply two pairs of byte values (one signed, one unsigned) into signed 16-bit values, and add them together all in one cycle. This requires us to ensure that input monomials are in $0, \ldots, 31$ and the coefficients in $-15, \ldots, 15$. We add together 32 results of VPMADDUSBW each time, which keeps the result between -32767 and 32766. We can then reduce the results again to numbers between 0 and 31. Arithmetic operations in extension fields over GF(31) are performed in straight schoolbook

form and do not use VPMADDUSBW instructions, because the sizes are not convenient for it.

Table 4 shows the running time of the signature generation and verification processes of our scheme for 80 and 128 bit quantum security. For comparison, we also provide here the running time of Gui [18]. Note again that the parameters of Gui were not chosen for quantum security. We expect the Gui parameters for 80 bit classical security to have about 62 bit quantum security. Similarly, the Gui parameters for 120 bit classical security provide 83 bit quantum security. Furthermore we want to emphasize that the implementation of [18] was far more optimized than ours (use of special processor instructions etc.). All the schemes listed in the table run on an Intel Xeon E3-1245 (Sandy Bridge) processor with 3.4 GHz.

Table 4. Comparison of the efficiency of HMFEv and Gui

Quantum security level (bit)		Sign. gen. time (ms)	Verification time (ms)
62	Gui (GF(2),96,5,6,6)	0.07	0.02
	Gui(GF(2),95,9,5,5)	0.18	0.02
	Gui(GF(2),94,17,4,4)	0.73	0.02
80	**HMFEv (GF(31),2,18,8)**	**0.131**	**0.0085**
	HMFEv (GF(256),3,9,12)	**0.261**	**0.0236**
83	Gui(127,9,4,6,2)	0.28	0.015
128	**HMFEv (GF(31),2,28,12)**	**0.259**	**0.0259**
	HMFEv (GF(256),3,15,16)	**0.443**	**0.063**

As the table shows, the performance of our scheme is at least comparable with that of Gui (note here that the Gui parameters provide significantly less security). Since, for increasing security level, the Gui parameters increase much faster than the parameters of our scheme, we believe that, for higher levels of security, our scheme will be much faster than Gui.

9 Conclusion

In this paper we proposed a new multivariate signature scheme called HMFEv which is obtained by applying the Vinegar modification to the MultiHFE scheme of Chen et al. [6]. By using this variation, we are able to reduce the number of components in the central map of the scheme and therefore to increase the efficiency significantly. We studied the security of our scheme against direct and rank attacks both theoretically and experimentally and showed that our scheme can not be attacked using differential methods or Hashimotos attack against the original MultiHFE scheme. We showed that our scheme is much more efficient

than the Gui and Rainbow signature schemes with regard to key and signature sizes.

Future work includes in particular the further optimization of the implementation to enable a better comparison of our results with those from [18] as well as a careful study on the effects of applying the Minus modification on HMFEv.

Acknowledgments. The third author is partially supported by NIST. The second and fourth authors would like to thank Academia Sinica for the second author's Investigator Award and Taiwan's Ministry of Science and Technology grant MoST-105-2923-E-001-003-MY3. We want to thank the anonymous reviewers for their valuable comments which helped to improve this paper.

Disclaimer. Certain algorithms and commercial products are identified in this paper to foster understanding. Such identification does not imply recommendation or endorsement by NIST, nor does it imply that the algorithms or products identified are necessarily the best available for the purpose.

References

1. Bernstein, D.J., Buchmann, J., Dahmen, E. (eds.): Post Quantum Cryptography. Springer, Heidelberg (2009)
2. Bettale, L., Faugère, J.C., Perret, L.: Cryptanalysis of HFE, multi-HFE and variants for odd and even characteristic. Des. Codes Cryptogr. **69**(1), 1–52 (2013)
3. Bogdanov, A., Eisenbarth, T., Rupp, A., Wolf, C.: Time-area optimized public-key engines: $\mathcal{MQ}$-cryptosystems as replacement for elliptic curves? In: Oswald, E., Rohatgi, P. (eds.) CHES 2008. LNCS, vol. 5154, pp. 45–61. Springer, Heidelberg (2008). doi:10.1007/978-3-540-85053-3_4
4. Cartor, R., Gipson, R., Smith-Tone, D., Vates, J.: On the differential security of the HFEv- signature primitive. In: Takagi, T. (ed.) PQCrypto 2016. LNCS, vol. 9606, pp. 162–181. Springer, Cham (2016). doi:10.1007/978-3-319-29360-8_11
5. Chen, A.I.-T., Chen, M.-S., Chen, T.-R., Cheng, C.-M., Ding, J., Kuo, E.L.-H., Lee, F.Y.-S., Yang, B.-Y.: SSE implementation of multivariate PKCs on modern x86 CPUs. In: Clavier, C., Gaj, K. (eds.) CHES 2009. LNCS, vol. 5747, pp. 33–48. Springer, Heidelberg (2009). doi:10.1007/978-3-642-04138-9_3
6. Chen, C.H.O., Chen, M.S., Ding, J., Werner, F., Yang, B.Y.: Odd-char multivariate Hidden Field Equations. IACR eprint (2008). http://eprint.iacr.org/2008/543
7. Daniels, T., Smith-Tone, D.: Differential properties of the *HFE* cryptosystem. In: Mosca, M. (ed.) PQCrypto 2014. LNCS, vol. 8772, pp. 59–75. Springer, Cham (2014). doi:10.1007/978-3-319-11659-4_4
8. Ding, J., Gower, J.E., Schmidt, D.S.: Multivariate Public Key Cryptosystems. Springer, New York (2006)
9. Ding, J., Yang, B.-Y.: Degree of regularity for HFEv and HFEv-. In: Gaborit, P. (ed.) PQCrypto 2013. LNCS, vol. 7932, pp. 52–66. Springer, Heidelberg (2013). doi:10.1007/978-3-642-38616-9_4
10. Ding, J., Hodges, T.J.: Inverting HFE systems is quasi-polynomial for all fields. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 724–742. Springer, Heidelberg (2011). doi:10.1007/978-3-642-22792-9_41
11. Ding, J., Schmidt, D.: Rainbow, a new multivariable polynomial signature scheme. In: Ioannidis, J., Keromytis, A., Yung, M. (eds.) ACNS 2005. LNCS, vol. 3531, pp. 164–175. Springer, Heidelberg (2005). doi:10.1007/11496137_12

12. Faugère, J.C.: A new efficient algorithm for computing Gröbner bases (F4). J. Pure Appl. Algebra **139**, 61–88 (1999)
13. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman and Company, New York (1979)
14. Hashimoto, Y.: Cryptanalysis of Multi HFE. IACR eprint (2015). http://eprint.iacr.org/2015/1160.pdf
15. Kipnis, A., Patarin, J., Goubin, L.: Unbalanced oil and vinegar signature schemes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 206–222. Springer, Heidelberg (1999). doi:10.1007/3-540-48910-X_15
16. Kipnis, A., Shamir, A.: Cryptanalysis of the HFE public key cryptosystem by relinearization. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 19–30. Springer, Heidelberg (1999). doi:10.1007/3-540-48405-1_2
17. Patarin, J., Courtois, N., Goubin, L.: QUARTZ, 128-bit long digital signatures. In: Naccache, D. (ed.) CT-RSA 2001. LNCS, vol. 2020, pp. 282–297. Springer, Heidelberg (2001). doi:10.1007/3-540-45353-9_21
18. Petzoldt, A., Chen, M.-S., Yang, B.-Y., Tao, C., Ding, J.: Design principles for HFEv- based multivariate signature schemes. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015. LNCS, vol. 9452, pp. 311–334. Springer, Heidelberg (2015). doi:10.1007/978-3-662-48797-6_14
19. Schwabe, P., Westerbaan, B.: Solving binary MQ with Grovers algorithm. https://cryptojedi.org/papers/mqgrover-20160901.pdf
20. Yang, B.-Y., Chen, J.-M.: Theoretical analysis of XL over small fields. In: Wang, H., Pieprzyk, J., Varadharajan, V. (eds.) ACISP 2004. LNCS, vol. 3108, pp. 277–288. Springer, Heidelberg (2004). doi:10.1007/978-3-540-27800-9_24

A Results of Our Computer Experiments with the Direct Attack Against HMFEv Systems over Small Fields

In this section we present the results of our computer experiments with the direct attack against HMFEv schemes over small fields. In particular, we wanted to answer the questions

1. Is the concrete choice of k and v (or only their sum) important for the degree of regularity of a direct attack against the scheme? and
2. Is the upper bound on d_{reg} given by Eq. (5) reasonably tight?

In order to answer the first question, we performed experiments of the following type: For fixed values of q and $s = k + v$, we varied the values of k and v. We then created the public systems of the corresponding HMFEv instances (for different values of ℓ) and solved these systems using the F_4 algorithm integrated in MAGMA. The experiments were (like all the experiments presented in this paper) performed on a server with 16 AMD Opteron cores (2.4 GHz) and 128 GB of RAM. However, as MAGMA is not parallelizable, our programs use only one core.

In our experiments, we fixed the field $\mathbb{F}$ to be GF(2) and the sum $s = k + v$ to be 9. We varied v in the interval $I = \{0, \ldots, 8\}$ and created HMFEv(GF(2), $s - v$, ℓ, v) instances (for increasing values of ℓ). After that, we fixed v of the variables to get a determined system and solved the resulting public systems by

the F_4 algorithm integrated in MAGMA. Table 5 shows, for $v \in I$, the highest degree of regularity we observed in these experiments. For each parameter set, we performed 10 experiments.

Table 5. Degree of regularity of HMFEv systems over GF(2) with $k + v = 9$

v	0	1	2	3	4	5	6	7	8
k	9	8	7	6	5	4	3	2	1
d_{reg}	3	4	4	5	5	5	5	5	4

As the experiments show, the concrete ratio between k and v has, as long as we choose v and k not too small, no influence on the degree of regularity of solving the public systems of HMFEv. For HMFEv schemes over larger fields the importance of the concrete choice of k and v decreases further, since those systems behave much more like random systems (see Sect. 6.2). We therefore choose, in order to increase the efficiency of our scheme, the parameter $k \in \{2, 3\}$ and increase v to reach the required level of security.

Is the Upper Bound on d_{reg} Given by Eq. (5) Reasonably Tight?

In order to answer this second question, we created for fixed values of q, k and v and varying values of ℓ public systems of HMFEv and solved them with the F_4 algorithm integrated in MAGMA. We increased the value of ℓ and therefore the numbers of equations and variables in the system until we reached the upper bound of (5) or ran out of memory.

It is obvious that we can only hope to find such systems for small field sizes. We therefore restricted to values of $q \in \{2, 3\}$.

By doing so, we identified the following "tight" instances of HMFEv

Scheme	Upper bound on d_{reg} (Eq. (5))	Experimental result
HMFEv(GF(2),1,ℓ,2)	3	3 for $\ell \geq 9 (n \geq 9)$
HMFEv(GF(2),2,ℓ,3)	4	4 for $\ell \geq 9 (n \geq 18)$
HMFEv(GF(2),3,ℓ,4)	5	5 for $\ell \geq 10 (n \geq 30)$
HMFEv(GF(3),1,ℓ,2)	5	5 for $\ell \geq 18 (n \geq 18)$

For most other HMFEv instances with $q \in \{2, 3\}$ and $k + v \leq 9$ we missed the upper bound given by Eq. (5) only by 1.

We believe that, also for these systems, we could have reached the upper bound given by Eq. (5) by increasing the parameter ℓ further. However, we did not have the necessary memory resources to solve HMFEv systems with more than 35 equations.